

EP 501 Final Exam Problem 2

Julio Guardado

December 11, 2019

Numerical Approaches to PDEs

Partial differential equations (PDEs) are equations that involve functions of multiple variables and the partial derivatives of those functions. The three main types of PDEs are elliptic, hyperbolic, and parabolic and they are classified as follows.

In the case where a PDE takes the form:

$$au_{xx} + bu_{xy} + cu_{yy} = 0, u = u(x, y) \quad (1)$$

its classification is determined by the value of the following relation:

$$b^2 - 4ac \quad (2)$$

This function is elliptic if this value is less than 0, hyperbolic if it is greater than 0, and parabolic if it equals 0.

Elliptic PDEs:

The canonical form of elliptic PDEs is called the Poisson equation and is as follows:

$$\nabla^2 u = f(x) \quad (3)$$

when $f(x)$ is 0, this equation becomes the Laplace equation. These equations can be used to model physical phenomenon such as electric potential and gravitational potential. For example, the solution to the equation $\nabla^2 \phi = -\rho/\epsilon_0$ describes the electric potential of the free space surrounding the charge density ρ . The analytical solution of these equations generally take the form of Fourier sine and cosine series and can be obtained through separation of variables.

In order to solve elliptic equations numerically, a finite difference approach is necessary. By representing the below equation (Laplace equation) with centered, 2nd order accurate finite differences, a system of equations can be made that will lead to a solution when solved using linear algebra solvers.

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (4)$$

let:

$$\left[\frac{\partial^2 \Phi}{\partial x^2} \right]_{i,j} = \frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta x^2} \quad (5)$$

and

$$\left[\frac{\partial^2 \Phi}{\partial y^2} \right]_{i,j} = \frac{\Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta y^2} \quad (6)$$

by substituting these relations into equation (6) and grouping like terms, the following system of equations emerges:

$$\Phi_{i+1,j} \left[\frac{1}{\Delta x^2} \right] + \Phi_{i-1,j} \left[\frac{1}{\Delta x^2} \right] + \Phi_{i,j+1} \left[\frac{1}{\Delta y^2} \right] + \Phi_{i,j-1} \left[\frac{1}{\Delta y^2} \right] + \Phi_{i,j} \left[\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2} \right] = 0 \quad (7)$$

In matrix form, this equation becomes

$$\underline{\underline{M}}\Phi = \underline{b} \quad (8)$$

where $\underline{\underline{M}}$ is a diagonally dominant matrix containing the various coefficients of $\Phi_{i,j}$. This system can be solved using any linear algebra solver, such as Gaussian elimination or Jacobi iteration. A consideration when performing these operations in MATLAB is the speed in which the program finishes calculating the solution. In order to increase efficiency, the use of sparse storage for $\underline{\underline{M}}$ is recommended.

This matrix only contains the interior coefficients for $\Phi_{i,j}$ and must be separately defined at the boundaries. Typical boundary conditions for elliptic PDEs include Dirichlet and Neumann conditions. Dirichlet boundary conditions are those where the value of the function at the boundaries is defined explicitly. Neumann conditions are those where the value of the derivatives is given at the boundaries. These conditions can be mixed and the value of the derivative can be given at the front boundary while the value of the function is known at the back boundary.

Hyperbolic Equations:

The canonical form of Hyperbolic PDEs is the wave equation, seen below.

$$u_t = -\alpha u_x \quad (9)$$

This equation can be used to model the propagation of a wave through a medium at a specific velocity. An example of this is the propagation of an electromagnetic wave. In this case the hyperbolic equation takes the form

$$\frac{\partial^2 E}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 E}{\partial t^2}$$

where E is the Electric field and c is the speed of light in the medium in which the wave is propagating. The analytical solution of this type of equations typically results in complex exponential and sinusoidal terms.

In order to solve these equations numerically, an analysis of stability of previous methods must be made. If attempting to use a forward time center space (FTCS) solution, a specific condition must be met. This condition can be derived by checking the growth of a single Fourier mode, $e^{ik_m \Delta x}$. By plugging this mode in for f_i^n , the following condition for the stability is obtained.

The system is stable if

$$\left| 1 - i \frac{v \Delta t}{2 \Delta x} \sin(k_m \Delta x) \right| \leq 1$$

However, this condition can never be met, and therefor the FTCS method is unconditionally unstable for hyperbolic equations. This is not the case, however, for the forward time, backward space (FTBS) method. A similar analysis can be done using a FTBS update formula (shown below) to achieve a condition for the stability of the system.

FTBS update formula:

$$f_i^{n+1} = f_i^n - \frac{v \Delta t}{\Delta x} (f_i^n - f_{i-1}^n)$$

Stability condition:

$$\frac{v \Delta t}{\Delta x} \leq 1$$

However, this stability is ruined when $v < 1$. In this case, a forward time, forward space update method can be applied and be stable under the same conditions. By combining these two update formulas, Godunov's method can be obtained. This method employs the concept of upwinding to ensure the solution remains stable. Upwinding involves the use of data "behind" the flow to compute the derivative. This results in the following update formula:

if $v < 0$

$$f_i^{n+1} = f_i^n - \frac{v \Delta t}{\Delta x} (f_{i+1}^n - f_i^n)$$

if $v > 0$

$$f_i^{n+1} = f_i^n - \frac{v \Delta t}{\Delta x} (f_i^n - f_{i-1}^n)$$

Another conditionally stable method of solving hyperbolic equations is the Lax-Wendroff Method. In order to derive this method, the Lax-Fredrich method must first be discussed. The Lax-Fredrich method employs an FTCS update formula and stabilizes it by introducing an artificial viscosity (averaging of adjacent cells) in the system. This artificial viscosity is a form of numerical diffusion. Numerical diffusion is the tendency for a solution to diffuse in the absence of a true physical viscosity term. This occurs due to truncation errors in finite difference formulas. Although it is a source of error, it allows for the use of FTCS solutions if absolutely necessary. The update formula for the Lax-Fredrich method is as follows:

$$f_i^{n+1} = \frac{1}{2} (f_{i+1}^n + f_{i-1}^n) - \frac{v\Delta t}{2\Delta x} (f_{i+1}^n - f_{i-1}^n) \quad (10)$$

This equation was expanded upon to create the Lax-Wendroff method for solving hyperbolic equations. This method employs centered half step updates, both in time and space, to compute the solution. This is beneficial as it allows for a second order accurate solution in space and time for linear advection equations. The update formula is as follows:

$$f_i^{n+1} = f_i^n + \frac{v\Delta t}{2\Delta x} (f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n+1/2}) \quad (11)$$

where

$$f_{i-1/2}^{n+1/2} = \frac{1}{2} (f_i^n + f_{i-1}^n) - \frac{v\Delta t}{2\Delta x} (f_i^n - f_{i-1}^n) \quad (12)$$

and

$$f_{i+1/2}^{n+1/2} = \frac{1}{2} (f_{i+1}^n + f_i^n) - \frac{v\Delta t}{2\Delta x} (f_{i+1}^n - f_i^n) \quad (13)$$

Both the Lax-Wendroff method and the Godunov method require initial conditions to begin iterating.

Parabolic Equations:

The canonical form for parabolic equations is of the form

$$u_t - \lambda u_{xx} = 0 \quad (14)$$

This equation is known as the diffusion equation and it describes diffusion through a medium with a diffusivity λ . An example of an equation in this form is the heat equation, seen below. This equation is used to model the diffusion of heat through a medium with thermal diffusivity λ .

$$\frac{\partial T}{\partial t} - \lambda \frac{\partial^2 T}{\partial x^2} = 0 \quad (15)$$

This equation requires initial and boundary conditions to solve and has an analytical solution that includes a combination of exponential and sinusoidal terms. This type of equation can be solved using both explicit and implicit finite difference equations.

Implicit Method:

A FTCS update formula can be developed using a similar method to the methods used previously. Using a forward euler derivative for time and a centered 2nd euler derivative for space, the following update formula can be obtained:

$$T_i^{n+1} = T_i^n + \lambda \frac{\Delta t}{\Delta x} (T_{i+1}^n - 2T_i^n + T_{i-2}^n) \quad (16)$$

This method is very straight forward and is stable when $\Delta t \leq \frac{\lambda}{2} \Delta x^2$. From this condition, it is clear that fine space grids will require a large amount of small time steps to achieve stability.

Explicit Methods:

In order to obtain an explicit method for solving parabolic PDEs, it is necessary to use backward time, centered space euler differences. After substituting these euler differences into the original formula, the following equation emerges:

$$T_{i-1}^{n+1} \left(\frac{-\lambda}{\Delta x^2} \right) + T_i^{n+1} \left(\frac{1}{\Delta t} + \frac{2\lambda}{\Delta x^2} \right) + T_{i+1}^{n+1} \left(\frac{-\lambda}{\Delta x^2} \right) = \frac{T_i^n}{\Delta t} \quad (17)$$

This can be expressed in matrix notation as $\underline{MT} = \underline{b}$ where \underline{M} is a diagonally dominant matrix containing the coefficients of T_i^n , \underline{T} is the vector containing the solution, and \underline{b} is the right hand side of the equation. This system can be solved using any linear algebra solver, such as Jacobi iteration or Gaussian elimination. An extension of this system of equations is the Crank-Nicholson method, also known as the trapezoidal method. This method uses a second order, centered in time finite difference. The resulting system of equations is as follows:

$$T_{i-1}^{n+1} \left(\frac{-\lambda}{2\Delta x^2} \right) + T_i^{n+1} \left(\frac{1}{\Delta t} + \frac{\lambda}{\Delta x^2} \right) + T_{i+1}^{n+1} \left(\frac{-\lambda}{2\Delta x^2} \right) = \frac{T_i^n}{\Delta t} - \frac{\lambda}{2} \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \quad (18)$$